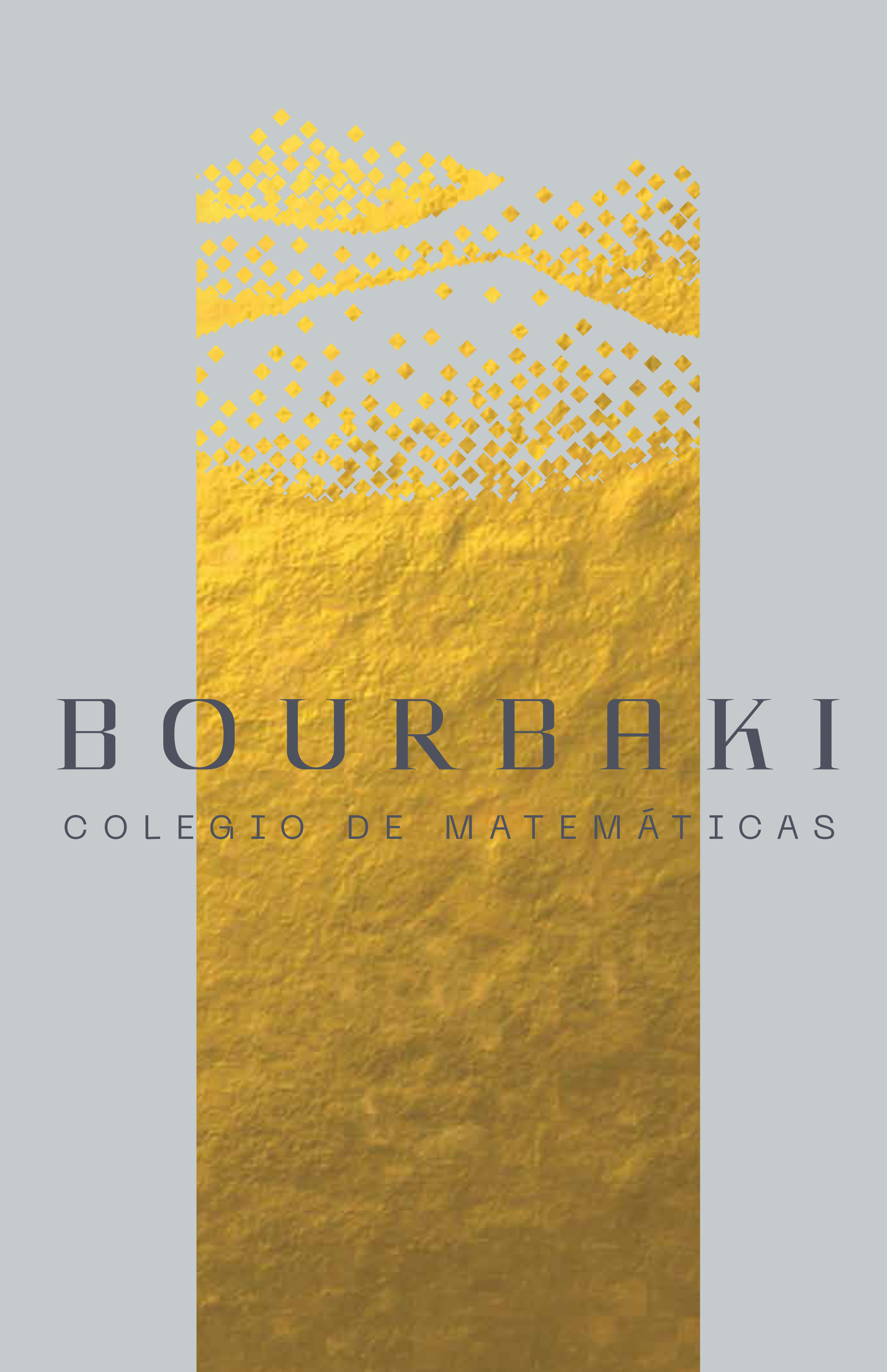

BOURBAKI
COLEGIO DE MATEMÁTICAS

BOURBAKI
COLEGIO DE MATEMÁTICAS

Índice

Introducción

Uno de los libros más bonitos de divulgación matemática es el escrito por el profesor Ian Stewart llamado "In Pursuit of the Unknown: 17 Equations That Changed The World".

Hemos decidido honrar aquella elección de la manera más modesta posible con nuestra recopilación de algunas ecuaciones que han revolucionado al aprendizaje automático mediante bases de datos.

Las ecuaciones que hemos elegido son las siguientes:

1. Perceptrón multi-capa y redes neuronales.
2. Entropía y entropía cruzada para distribuciones de probabilidad.
3. Fórmula de Bayes e inferencia bayesiana.


ÍNDICE



Les damos la bienvenida a nuestro mini-curso "17 ecuaciones que cambiaron Machine Learning". El objetivo de este curso es presentar algunas de las ecuaciones más importantes de machine learning que enseñaremos a lo largo de la Ruta a través de la Ciencia de Datos.

La presentación que hemos elegido para este mini-curso incluye tanto la relación que existe entre cada ecuación y su uso en machine learning, así como una explicación del contexto histórico en el que se desarrolló. A lo largo de los distintos cursos de nuestra Ruta a través de la Ciencia de Datos los estudiantes van a profundizar tanto en el significado como en las sutilezas de estas ecuaciones. Este mini-curso debe entenderse como un panorama general sobre el área. La presentación que hemos elegido es progresiva respecto a cómo encontrarán estos temas en nuestros cursos.

1. (Perceptrón multi-capa y redes neuronales, 1943)

   El perceptrón con una sola capa es un modelo matemático que realiza predicciones por medio de un promedio ponderado de las características de nuestros datos. Por ejemplo, el rendimiento de una compañía podría depender en cierto porcentaje de sus ventas a mayoristas y en otro porcentaje de sus ventas a minoristas, y estos porcentajes no siempre son 50 %. En este caso diremos que nuestra variable se comporta linealmente respecto a otras, pero en algunos casos un fenómeno podría no comportarse linealmente. La ecuación del perceptrón multi-capa propone un comportamiento no-lineal por medio de concatenación y

composición de perceptrones multi-capa. Las poderosas redes neuronales profundas que se utilizan en tan diversas áreas de la ciencia de datos utilizan estas fórmulas para aproximar mejor a las variables.

$$\ldots \rho_{i+1}(W_{i+1}\rho_i\left(W_i X + W_i'\right) + W_{i+1}')\ldots$$

Podrán encontrar más información sobre estos temas en nuestros cursos Machine Learning and AI for the Working Analyst y Matemáticas para la Ciencia de Datos.

2. (Entropía y entropía cruzada para distribuciones de probabilidad, 1948)

Supongamos que D es una muestra de nuestra población que satisface ciertas características fijas. Por ejemplo, podríamos pensar en características demográficas. Si deseamos segmentar a esta población en n grupos distintos, el cálculo de la entropía nos permite valorar la importancia de estas características.

Denotaremos por p_i al porcentaje de la población D, que además pertenece a cada una de las clases. La entropía dice lo siguiente:

$$E(D) = -p_1 log_2\left(p_1\right) - \ldots - p_n log_2\left(p_n\right)$$

Cuando esta cantidad es pequeña esto significa que las características que determinan a D son suficientemente representativas para la segmentación que deseamos.

Podrán encontrar más información sobre estos temas en nuestros cursos Machine Learning and AI for the Working Analyst y Especialización en Deep Learning.

3. (Fórmula de Bayes e inferencia bayesiana, 1763)

Supongamos que D corresponde a los datos que serán nuestra evidencia, mientras que M será un modelo matemático que intenta aproximar el fenómeno que estamos investigando. Por ejemplo, podríamos pensar en D como un histórico de nuestros clientes junto a sus características y la cantidad que se les ha facturado hasta la fecha. El modelo M en este caso podría ser uno que elige las características más importantes de nuestros clientes (feature selection).

La fórmula de Bayes dice lo siguiente:

$$\mathbb{P}(M|D) = \mathbb{P}(D|M) \cdot \frac{\mathbb{P}(M)}{\mathbb{P}(D)}$$

En algunos casos, el lado derecho de la ecuación es sencillo de calcular e inclusive podemos añadir conocimiento de experto sobre el problema para reducir un poco el espacio de búsqueda. En el caso de los clientes, significa que es posible actualizar la probabilidad sobre el feature selection a medida que nuestras bases de datos crecen, e inclusive añadir información de negocio del estilo: solo me interesan cierto número de características. La entropía cruzada permite calcular la diferencia entre dos hipótesis D y F sobre la importancia de estas caracte-

rísticas.

Podrán encontrar más información sobre estos temas en nuestros cursos Machine Learning and AI for the Working Analyst y Matemáticas para la Ciencia de Datos.

4. (Márgenes entre clases y máquinas de soporte vectorial, 1992)

Supongamos que D es una base de datos que contiene información clínica sobre pacientes que tienen o no una enfermedad. Deseamos encontrar un modelo matemático M que identifique los patrones que causan esta enfermedad. Un enfoque clásico en machine learning es buscar aquel modelo que cometa la menor cantidad de errores en nuestra base de datos D. Las máquinas de soporte vectorial proponen una manera distinta para encontrar estos patrones: concentrarse en buscar aquel modelo que se aleje simultáneamente de los registros que están enfermos y de los que no lo están.

$$M_{SVM} = \underset{M}{argmax}(Marg(M,S))$$

La cantidad $Marg(M,S)$ define el margen entre la base de datos y la frontera de decisión del modelo. Esta poderosa idea tiene muchas ventajas desde un punto de vista estadístico.

Podrán encontrar más información sobre estos temas en nuestros cursos Machine Learning and AI for the Working Analyst y Matemáticas para la Ciencia de Datos.



5. (Descomposición del Error cuadrático medio como trade-off entre varianza y sesgo, 1805, 1991)

   Supongamos que D es una base de datos que incluye información sobre transacciones en línea, así como el monto de la transacción. Si M es un modelo que busca predecir el monto utilizando a las características, podemos calcular el Error cuadrático medio de la siguiente manera:

$$Error\,(M,D) = \frac{1}{N} \cdot \sum_{i \leq N} ((M(x_i) - y_i)^2 = Var(M) + Bias(M)$$

   Para el caso en el que supongamos que nuestro modelo M es lineal, es decir, cuando la predicción es un promedio ponderado de las características de cada transacción, entonces podemos interpretar al primer sumando como la varianza de los pesos durante el entrenamiento y al segundo como el ruido el cual tradicionalmente es gaussiano. Esta descomposición nos permite comprender el dilema entre el ajuste y el sobre-ajuste de los modelos en ciencia de datos.

   Podrán encontrar más información sobre estos temas en nuestros cursos Matemáticas para la Ciencia de Datos.

6. (Funciones de activación sigmoide y soft-max, 1858)

   Uno de los problemas más difíciles en ciencia de datos es el de encontrar una explicación a los modelos matemáticos entrenados con bases de datos que se pueda traducir fácilmente en términos simples y de preferencia amigables con los usuarios. Ya hemos hablado de los

modelos lineales, los cuales tienen grandes ventajas en este sentido, aunque otras desventajas como un gran error de aproximación para fenómenos más complejos y en particular no-lineales. Una de las herramientas más poderosas para resolver este problema son las funciones de activación conocidas como sigmoides o soft-max, las cuales son la base de las regresiones logísticas y nos permiten explicar con transparencia el origen de una predicción.

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Podrán encontrar más información sobre estos temas en nuestros cursos Machine Learning and AI for the Working Analyst y Especialización en Deep Learning.

7. (Regularizador de Tychonov y métodos dispersos, 1970)

Supongamos que una base de datos D contiene las reseñas de algún producto que vendemos en nuestra compañía y deseamos construir un modelo de machine learning que prediga cuáles son las reseñas positivas y cuáles son las negativas. La cantidad de palabras distintas que aparecen en esta base de datos podría crecer exponencialmente con el tamaño n de la base de datos y por ello es necesario eliminar algunas de las palabras para encontrar un mejor modelo. Las técnicas de regularización utilizan métricas que permiten reducir el número de palabras consideradas por el modelo de machine learning M. En el caso de Ty-



chonov, la fórmula es la siguiente:

$$Error_{Ridge}(M, D) = \frac{1}{N} \cdot \sum_{i \leq N} ((M(x_i) - y_i)^2 + \lambda ||M||_2$$

La ecuación anterior corresponde a la regularización Tychonov o Ridge y al modificar la última parte de la fórmula introducimos otras técnicas como los métodos dispersos tipo lasso, entre otros. En este caso estamos utilizando el error cuadrático medio, pero también es posible hacerlo mediante otras métricas.

Podrán encontrar más información sobre estos temas en nuestros cursos Machine Learning and AI for the Working Analyst, Matemáticas para la Ciencia de Datos y Especialización en Deep Learning.

8. (Ley de los grandes números para procesos estacionarios, 1539)

Si suponemos que $X_1, X_2, \ldots, X_N$ es una familia de observaciones sobre algún fenómeno, diremos que esta familia es estacionaria cuando la distribución de cualesquiera dos subconjuntos de estas observaciones con la misma distancia en sus índices es la misma. Intuitivamente, estamos diciendo que las observaciones no solo corresponden al mismo fenómeno, sino que además, al considerarlas en grupo, están correctamente organizadas.

La ley de los grandes números para estos fenómenos predice la existencia de un límite:

$$\lim_{N\to\infty}\left(\frac{X_1(m)+\ldots+X_n(m)}{N}\right)=L$$

Dentro de los procesos estacionarios están tanto el ruido blanco como las cadenas de Markov e incluso los procesos ARIMA para series de tiempo. Estos casos corresponden a las siguientes intuiciones sobre la base de datos: muestreos estadísticamente representativos, procesos con memoria de corto plazo y series de tiempo sin tendencia o temporalidad, respectivamente.

Podrán encontrar más información sobre estos temas en nuestros cursos Machine Learning and AI for the Working Analyst, Matemáticas para la Ciencia de Datos y Especialización en Deep Learning.

9. (Distribución gaussiana y el teorema central de Lévy, 1920)

Supongamos que tenemos una base de datos D que contiene d características, todas ellas continuas, tal que el vector de sus promedios es igual a μ mientras que su matriz de covarianza es S. La distribución gaussiana que mejor aproxima a esta base de datos es la definida por la siguiente fórmula:

$$\mathbb{P}_{Gauss}((-\infty,x_1]\times\ldots\times(-\infty,x_d])=\frac{1}{(2\pi)^{d/2}|S|^{d/2}}\int_{-\infty}^{x_1}\ldots\int_{-\infty}^{x_d}e^{-\frac{1}{2}(t-\mu)S^{-1}(t-\mu)^T}dt$$

Aunque es plausible que una base de datos no sea correctamente apro-

ximada por una distribución gaussiana, el teorema del límite central asegura que los errores que comete la aproximación de la ley de los grandes números en el punto anterior siempre serán gaussianos. Este teorema permite definir intervalos de confianza, los cuales están sujetos al cumplimiento de las hipótesis sobre la base de datos.

Podrán encontrar más información sobre estos temas en nuestros cursos Machine Learning and AI for the Working Analyst y Matemáticas para la Ciencia de Datos.

10. (Factorización de matrices en valores singulares, 1873)

Pensemos en el caso de Netflix y su base de datos, donde para cada usuario conocemos las películas o series que han visto hasta el momento. El principal problema de esta base de datos es que la gran mayoría de los registros tienen menos del uno porciento de registros nulos, por lo que es imposible matemáticamente hablando encontrar semejanzas entre ellos. Gracias a los teoremas de factorización de matrices es posible descomponer la base de datos de Netflix en tres bases de datos con tamaños mucho menores que no solo permiten calcular semejanzas entre usuarios, sino también entre películas e incluso entre películas y usuarios, lo cual es muy útil para construir sistemas de recomendación.

$$X_{Netflix} = U_{Usuarios} \cdot D \cdot V_{Series}$$

Podrán encontrar más información sobre estos temas en nuestro curso Matemáticas para la Ciencia de Datos.

11. (Algoritmo del gradiente descendente, 1847)

    El proceso del entrenamiento de los modelos utilizando bases de datos es un proceso complicado para el cual es necesario tener algoritmos que aproximen a nuestros datos eficazmente. Una de las grandes ideas matemáticas utilizadas en machine learning fue propuesta por Cauchy como método de optimización de las funciones de error. La idea intuitiva detrás de este algoritmo es la siguiente: si un modelo comete un error en nuestro conjunto de entrenamiento y calculamos la derivada de este error, al restarle iterativamente esta derivada estamos disminuiremos el error en las siguientes iteraciones.

$$M_{t+1} = M_t - \nabla Error\,(M_t)$$

    Podrán encontrar más información sobre estos temas en nuestros cursos Matemáticas para la Ciencia de Datos y Especialización en Deep Learning.

12. (Ecuaciones de Bellman y aprendizaje por refuerzo, 1953)

    Supongamos que estamos en una posición s en un juego de ajedrez parametrizada por distintas variables (posiciones de las piezas, número de jugadas, etc.). Si π es una estrategia de juego, es posible evaluar con exactitud la calidad de esta estrategia en s utilizando las evaluaciones

de otras estrategias.

$$V_\pi(s) = \mathbb{E}[R(s,\pi(s))] + \gamma \cdot \sum_{s'} \left(\mathbb{P}\left(s'|s,\pi(s)\right) V_\pi\left(s'\right)\right)$$

Gracias a esta ecuación es posible reducir el espacio de búsqueda donde encontramos las estrategias óptimas y así vencer lo que se conoce como la maldición de la dimensión. Estas ecuaciones han permitido construir modelos de inteligencia artificial verdaderamente poderosos.

Podrán encontrar más información sobre estos temas en nuestros cursos Machine Learning and AI for the Working Analyst, Matemáticas para la Ciencia de Datos y Especialización en Deep Learning.

13. (Cross-correlation o convoluciones entre tensores, 1807)

Supongamos que tenemos una imagen I a la cual queremos aplicarle un filtro F. Los objetivos de hacer esto podrían ser muy diversos, pero una de las intuiciones más útiles es porque queremos comprimir la información. Para este caso y muchos otros como la detección de objetos, por ejemplo, las correlaciones son operaciones matemáticas muy importantes que promedian los píxeles de una imagen siguiendo una regla constante a lo largo y alto de la imagen. La fórmula de la correlación cruzada es la siguiente:

$$(I \star F)_{r,s} = \sum_{i\leq N} \sum_{j\leq N} \left(F\left(i,j\right) I_{r+i,s+j}\right)$$

Podrán encontrar más información sobre estos temas en nuestros cursos Machine Learning and AI for the Working Analyst y Especialización en Deep Learning.

14. (Función de pérdida adversarial para modelos generativos, 2014)

Las tareas generativas dentro de machine learning son muy complicadas, pues requieren aproximar funciones. Al tratar inteligencia artificial estas funciones se vuelven muy complejas, como el caso de las imágenes, en donde tenemos tantos grados de libertad como píxeles. Una manera muy inteligente para optimizar a los modelos generativos es mediante la ayuda de un modelo de clasificación supervisado. La idea general detrás de las Generative Adversarial Networks consiste en simultáneamente construir una base de datos donde, por definición, los registros generados son falsos y los de una base de datos fija serán los únicos verdaderos. La función de optimización en este caso coincide con un juego de suma cero, donde lo que gana un modelo de clasificación es lo que pierde el modelo generativo y viceversa.

$$\underset{G}{max}(\underset{C}{min}(err_C(D))$$

Podrán encontrar más información sobre estos temas en nuestro cuso de Especialización en Deep Learning.

15. (Memoria de largo plazo y modelos recurrentes, 1999)

Así como las convoluciones son una operación matemática que per-



mite inducir un sesgo sobre las operaciones entre imágenes, el procesamiento del lenguaje natural y las series de tiempo requieren que las redes neuroanles tengan mejores arquitecturas que se amolden a la estructura de los datos. Las redes neuronales recurrentes modernas proponen un método para que las memorias de largo y corto plazo simpaticen y se puedan encontrar correlaciones largas y cortas al mismo tiempo. Semánticamente, lo anterior es muy importante, pues tanto las palabras lejanas como las cercanas pueden tener alguna importancia en nuestros textos.

$$l_{t+1} = f(p_t, c_{t-1}) \odot l_t + i(p_t, c_{t-1}) \odot a_t$$

Podrán encontrar más información sobre estos temas en nuestros cursos Machine Learning and AI for the Working Analyst y Especialización en Deep Learning.

16. (Regla de la cadena y backpropagation, 1986)

Las redes neuronales profundas pueden tener incluso trillones de parámetros que se entrenarán mediante un algoritmo conocido como backpropagation. Este algoritmo hace una elección inteligente y eficaz del cálculo de las derivadas que se harán para implementar el método del gradiente. Recordemos desde la fórmula de los perceptrones, pero también para el caso de las redes convolucionales y recurrentes, que las iteraciones entre distintas capas buscan aumentar la capacidad expresiva

de los modelos, lo cual no tiene que ser necesariamente fácil desde un punto de vista computacional. La fórmula en la que se basa este algoritmo es la famosa regla de la cadena, la cual relaciona, por ejemplo, un portafolio de inversión P con los combustibles que se utilizan para generar bienes B_i de los cuales dependerá el portafolio P.

$$\frac{\delta P}{\delta C} = \sum_{j \leq n} \left(\frac{\delta C}{\delta B_j} \cdot \frac{\delta B_j}{\delta X}\right)$$

Podrán encontrar más información sobre estos temas en nuestros cursos Matemáticas para la Ciencia de Datos y Especialización en Deep Learning.

17. (Mecanismo de atención y modelos semi-supervisados, 2017)

Supongamos que deseamos que un modelo de inteligencia artificial aprenda a buscar información dinámicamente utilizando algunos casos en los que buscó información exitosamente. Existe una arquitectura que permite realizar esto y es conocido como el mecanismo de atención, la ecuación fundamental la mostramos a continuación y en este caso podemos entender a B como aquello que estamos buscando en la memoria M mientras que I serán las instrucciones dinámicas que aprenderá el modelo de inteligencia artificial. En el caso del texto esta ecuación es muy importante pues permite realizar búsquedas en partes anteriores de nuestros textos.



$$Atencion(B, I, M) = Softmax(\frac{B \cdot I}{\sqrt{d}}) \cdot M$$

Podrán encontrar más información sobre estos temas en nuestro curso

Especialización en Deep Learning.



escuela-bourbaki.com

